

KidzCanCode

IGNITE, INSPIRE, INNOVATE

www.kidzcancode.com

Walking the Plank

KIDZCANCODE

TREVOR WARREN

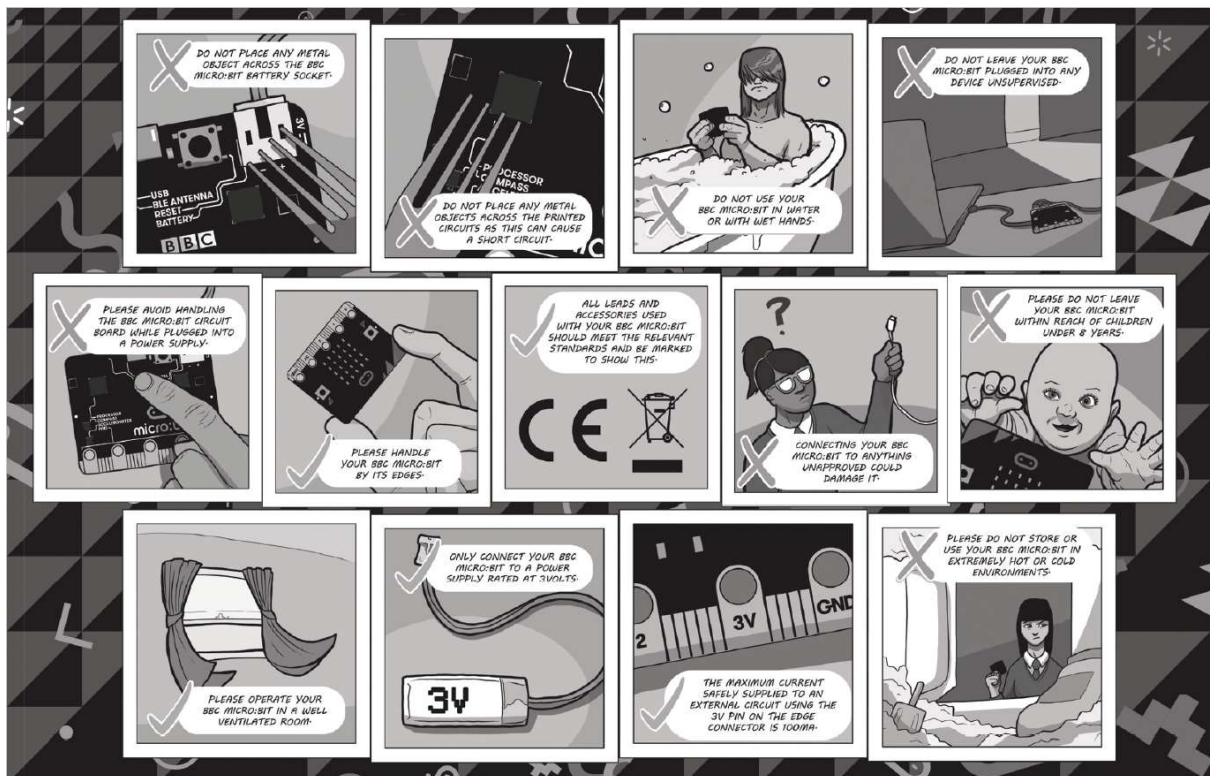
Tutorial 2.1a – Walking The Plank

1. Safety Warnings

THE MICRO:BIT IS AN EXPOSED BOARD, TO BE USED WITH CARE PLEASE RETAIN THIS INFORMATION FOR FUTURE REFERENCE. You can read the detailed document at - <http://microbit.org/guide/safety-advice/>

a. General Safety Warnings

Using the BBC micro:bit is easy to use but is designed to have all the electrical parts on display. This does mean there's a small risk that the parts can be damaged and even overheat with a risk of injury but a little bit of care and caution will ensure you and your micro:bit will stay fit and healthy.

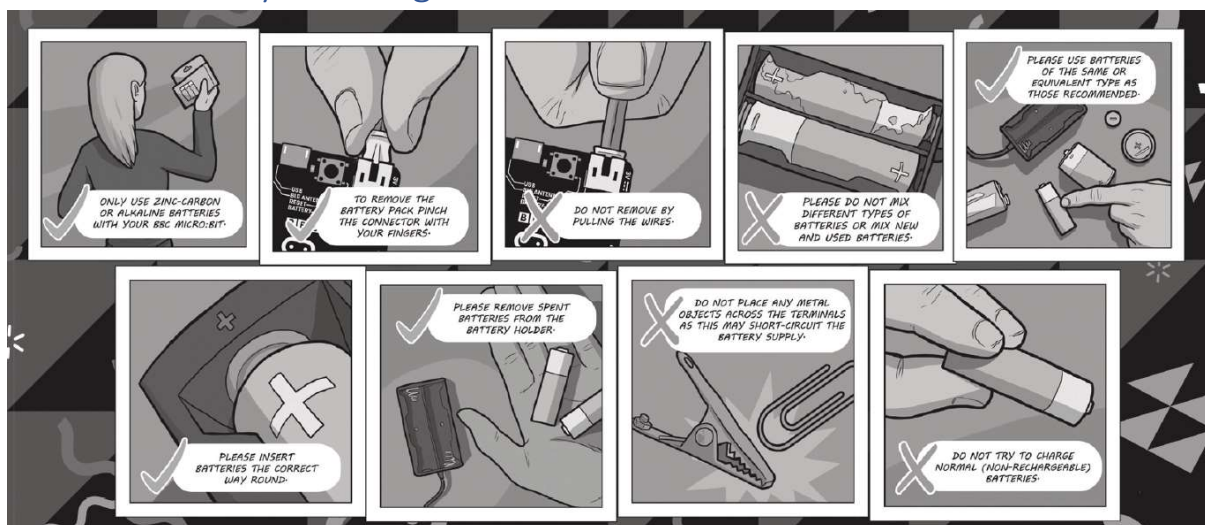


1. Always keep your BBC micro:bit in the anti-static bag when not in use. It's good practice for students to earth themselves before handling it.
2. Please handle your BBC micro:bit by its edges. This minimises the risk of damage through an electrostatic discharge.
3. Please use the battery pack and the USB lead provided to power your micro:bit. Do not use portable battery chargers or USB charging ports (often marked with a lightning bolt or 'SS'), to power your micro:bit. Using these may damage your micro:bit and stop it working properly.
4. Please avoid handling the BBC micro:bit circuit board while plugged into a power supply.
5. All peripherals (for example: USB cable, battery holder, sensors) used with your BBC micro:bit should comply with the relevant standards and should be marked accordingly.
6. Connecting your BBC micro:bit to any unapproved peripherals could damage your BBC micro:bit
7. Please do not attempt to keep using faulty micro:bits. If a school-issued micro:bit develops a fault, contact the vendor immediately.

Tutorial 2.1a – Walking The Plank

8. The maximum current safely supplied to an external circuit using the 3V pin on the edge connector is 100mA. Please make sure this limit is not exceeded.
9. Please do not store or use your BBC micro:bit in extremely hot or cold environments.
10. Do not place any metal objects across the printed circuits on the board as this can cause a short circuit damaging your BBC micro:bit. This can cause risk of burn or fire.
11. Do not use your BBC micro:bit in water or with wet hands.
12. Do not leave your BBC micro:bit plugged into a computer or any other device unsupervised.
13. Please do not leave your BBC micro:bit within reach of children under 8 years of age.
14. Please operate your BBC micro:bit in a well ventilated room To remove the battery pack, pinch the connector with your fingers. Do not remove by pulling the wires.

b. Battery Warnings



1. Do not try to charge normal (non-rechargeable) batteries
2. Please do not mix different types of batteries or mix new and used batteries.
3. Please use batteries of the same or equivalent type as those recommended.
4. Please insert batteries the correct way round (with the correct polarity).
5. Please remove spent batteries from the battery holder.
6. Do not short-circuit the battery supply terminals, for example by placing a metal object across the terminals.
7. Only use Zinc or Alkaline batteries with your BBC micro:bit.
8. Please do not use rechargeable batteries

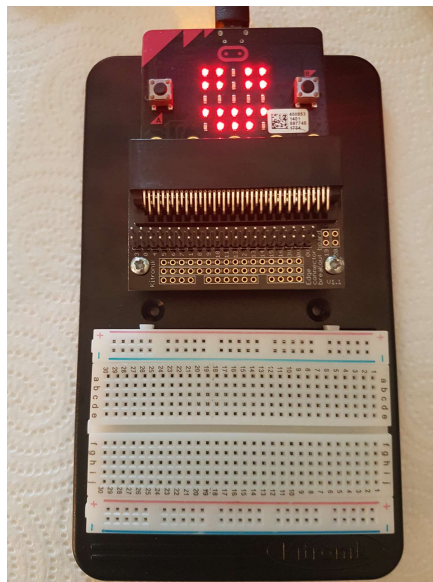
Tutorial 2.1a – Walking The Plank

2. Pre-requisites

If you have questions with the assembly of the micro:bit, edge connector breakout board, mounting board and the breadboard please drop us a note at help@kidzcancode.com. The edge connector board, mounting board and the breadboard are part of the Kitronix Inventors kit which needs to be purchased separately.

To be able to perform this tutorial you will need the following components –

1. Parts required –
 - a. 1 x BBC Micro:bit
 - b. 1 x Mounting Plate
 - c. 1 x Edge connector breakout board
 - d. 1 x Bread board
2. Assembly required –
 - a. Bread board mounted on top of the mounting plate
 - b. BBC Micro:bit inserted into the Edge Connector breakout board



Before proceeding please check your setup and confirm that all the required parts are configured as demonstrated in the above picture.

Tutorial 2.1a – Walking The Plank

3. Learning Objectives

The objectives of this tutorial are to introduce the student to the concepts of pitch and roll through programming the micro:bit. Through this tutorial the student will also learn how to build a simple interactive game using the input on the micro:bit to control aspects of the game. We also briefly touch upon the use of loops and implement a loop using the, “if then else” loop within our code blocks.

This tutorial builds upon concepts introduced in previous tutorials so please make sure you have covered the previous tutorials before you dive into this one. So overall this tutorial intends to build upon concepts learnt in previous tutorials while exploring new concepts.

In future tutorials we will continue to build upon the concepts learned here and will build more complex interactive games using the functionality provided by the micro:bit.

Explore the following concepts through making –

1. Programming the micro:bit using the makecode interface
2. Exploring the concepts of pitch and roll
3. Exploring the concepts of loops with a focus on the – if then else loop



Tutorial 2.1a – Walking The Plank

4. Activity

a. Activity

This activity involves creating an interactive game called “Walking The Plank”. The objective of this game is to walk around an agreed path (preferably all around the room, limiting yourself to a narrow path) without tilting the micro:bit board. Players place the board on the rear of the palm and walk all around the room.



Winners of the game are those who are able to walk all around the agreed track within the room without tilting the board. The micro:bit is programmed to detect any change in the level of the board i.e. any changes from its starting horizontal position. As soon as the board detects a change in position i.e. the board is no longer horizontal it flashes a message and ends the game.

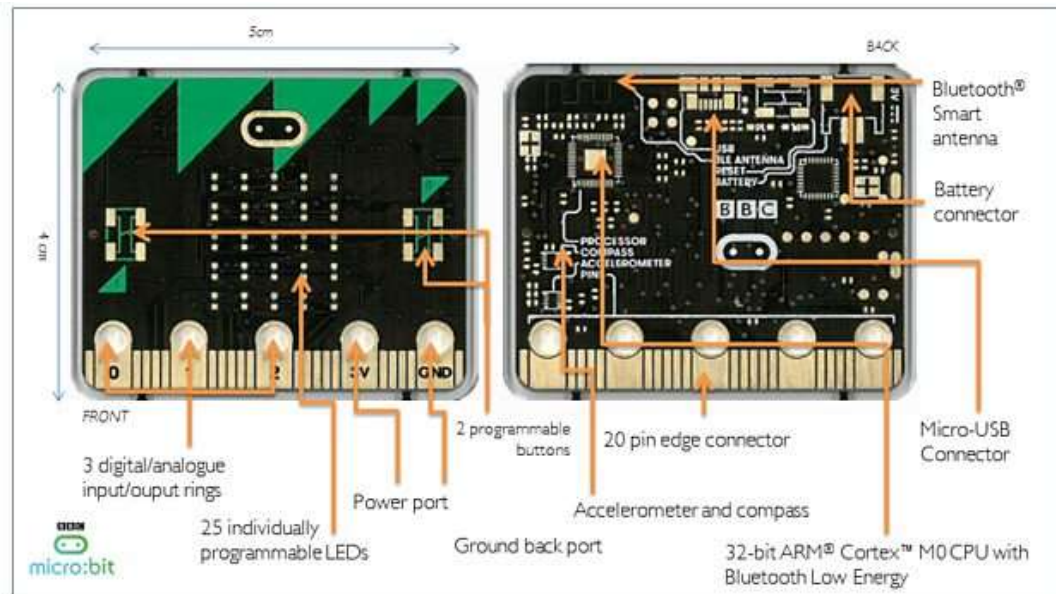
b. How Does It Work

This section talks about the BBC micro:bit and what it's made up of. If you have already read through this section then feel free to skip directly to the next section. The BBC micro:bit is a powerful little board and has various types of sensors on board. Here's what makes up the BBC micro:bit.

1. Size: approx. 5cm x 4cm.
2. Weight: 8g.
3. Processor: 32-bit ARM Cortex M0 CPU.
4. Bluetooth Low Energy.
5. Digital Compass.
6. Accelerometer.
7. Micro-USB controller.
8. 5x5 LED matrix with 25 red LEDs.
9. Pins for connecting external sensors, LED's, etc.

Here's what the micro:bit looks like –

Tutorial 2.1a – Walking The Plank



Front of the board (left hand side)

1. Button A (left button with edge connector at the bottom) – labelled A on the board
2. Button B (right button with edge connector at the bottom) – labelled B on the board
3. P0 (left large pin (crocodile clip port) with edge connector at the bottom) - labelled 0 on the board
4. P1 (middle large pin (crocodile clip port) with edge connector at the bottom) - labelled 1 on the board
5. P2 (right large pin (crocodile clip port) with edge connector at the bottom) - labelled 2 on the board
6. +3V - labelled 3V on the board. This is 3V PWR OUT
7. GND
8. P3 – P22 pins from left to right with edge connector at the bottom. Referred to as Pins when referencing that part of the board. Text will talk about 'pins' when referring to individual connections or the general way of connecting to the board – not labelled on the front of the board
9. LED matrix referred as the 'screen' - not labelled on the board
10. LED coordinates starting at 0,0 top left corner and ending at 4,4 at the bottom corner - not labelled on the board

The order of the large pins as follows: P0 P1 P2 3V GND labelled 0, 1, 2, 3V GND on the board

Rear of the board (Right hand side)

1. USB Plug (Micro-USB plug) – labelled USB on the board
2. Button R (reset button) – labelled Reset on the board
3. Status LED – not labelled on the board
4. Battery socket – labelled Battery on the board

Other components on the board include

Tutorial 2.1a – Walking The Plank

1. Accelerometer
2. Compass
3. Bluetooth Smart Technology Antenna
4. AAA Battery Holder - not labelled on the board
5. Processor (Cortex M0)

The BBC micro:bit is programmable in a few different languages. You can write code for the micro:bit using the Makecode block coding interface, Javascript, Python or even in C. Most of our tutorials will cover the use of the Makecode block coding interface built by Microsoft for the micro:bit.

The micro:bit board has many on-board sensors of which the accelerometer is one. The accelerometer sensor on board the micro:bit detects motion, speed and can be used to track changes in positioning of the board including changes in speed. The micro:bit sensors by itself will do nothing unless they are programmed (through code) to perform the relevant measurements. The game therefore consists of a set of instructions that involve performing relevant measurements, making decisions depending on the nature of the measurements and then displaying messages depending on the conditions encountered.



Tutorial 2.1a – Walking The Plank

5. Let's write some code

It's time to write some code and get going with coding our game. So let's head over to the micro:bit block code editor page (<https://makecode.microbit.org/>) and get coding!!!

a. On Start

In the following section we will dive into the code you will put together for this tutorial. The code is split into two separate sections just to make things a bit easier. The first code block below covers off concepts that you would have covered in previous tutorials so while we cover the relevant blocks, we will not dive into a lot of depth here.

In the first code block, we use the “on start” block provided by the micro:bit which is run only “once” during a given program. You can only trigger the “on start” block once again when you hit the re-set button or pull the power plug and reboot the board. So please do keep that in mind with regards to the “on start” block of code. You want to put stuff into the “on-start” code block that you want run at the start of the program.



- The “on start” block of code simply shows a string
- We then pause briefly to ensure that the reader has been able to see the text being displayed
- We then show a pre-defined icon
- We again pause briefly for a second

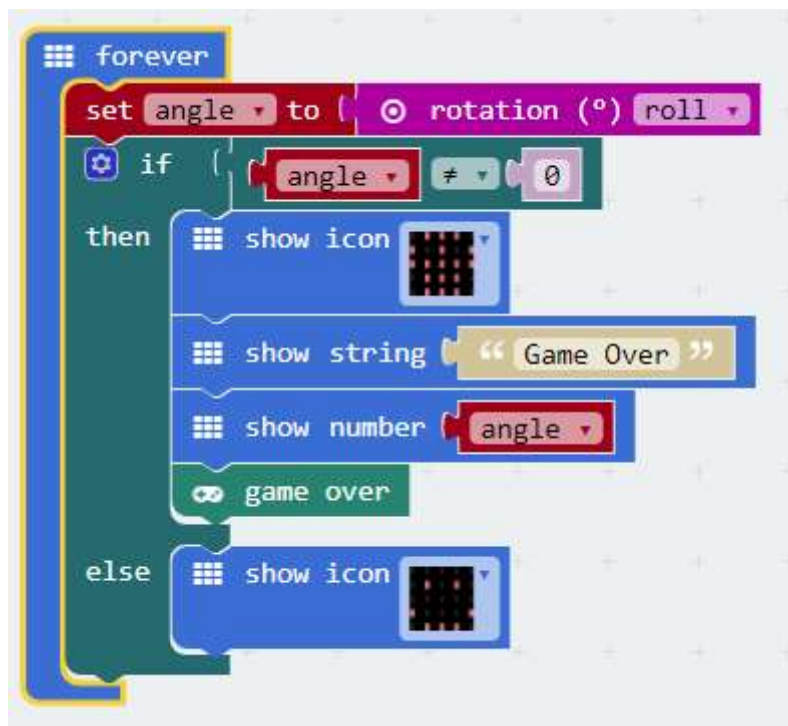
This brings the on-start block of code to an end. Feel free to dive in and customize the “on-start” code block with additional code that you want to put in. Please note that the addition of the “pause” commands (similar to our use of the wait commands in scratch) is intended to inject wait and slow down processing so that the flow of the program makes sense to the human being. To the computer, not having the wait command just lets it breeze through all the code one instructions after another.

Tutorial 2.1a – Walking The Plank

b. Forever Code Block

The next block of code deals with the main content where we create our “Walk The Plank” game.

- In this section we use the “forever” block which runs code repeatedly until the board is reset or the power is turned off.
- We start off by using the “set angle to” code block which basically measures the current angle the board is placed in.
- The “set angle” code block is assigned a value which is the current “rotational roll” of the board which is obtained by using the “rotation (o) roll” code block.
- Be careful cause you rotation has two options, pitch and roll. This game requires use of the roll option.
- We then set up a decision loop using the if-then-else block. The if-then-else code block is used to determine if the board is horizontal i.e. angle = 0, or if the board is not longer level (horizontal) in which case the angle will be > 0 (This can also be negative, but let’s not dive into that detail here. We’ll go into that later).
- If we detect that the board is not level anymore we display Game over text and terminate the game.

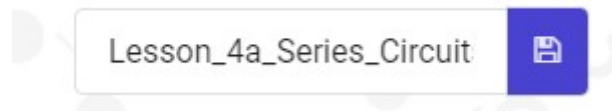


You are encouraged to make changes, improvise and customize the game using your own ideas.

Tutorial 2.1a – Walking The Plank

6. Downloading Your Code To The micro:bit

Once you have completed the program, enter a name for your program using the option provided below i.e. in the text box adjacent to the small save button.



Now that you have given your program a name and saved it you can download it your micro:bit. But before we do that let's confirm what drive your micro:bit shows up as. On most machines the micro:bit will show up as an additional USB drive. So head over into windows explorer and confirm what drive name (D:, E:, F:, G:, etc.) the micro:bit shows up as. You need to absolutely be sure what drive the micro:bit shows up as. Once you've confirmed what drive the micro:bit shows up as on your machine you can select the right drive when downloading the code to the micro:bit. If in doubt please ask the volunteer/mentor/session facilitator helping out.



To download the newly written code to the micro:bit, hit the download button shown above. You should now see a dialog box open up and you will be asked to save the file somewhere on your machine. Please choose the drive your micro:bit shows up as i.e. D: or E: or F: or whatever it shows up as on your machine. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

If hitting the download button shown above does not open up a dialog box asking you to save to the micro:bit please save the file (you will have a <filename.hex> file) to your desktop. Then open up windows explorer and drag that file onto the drive which is your micro:bit. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

Please feel free to customize the code blocks, have a play. Add your own custom code and re-download the code to the micro:bit. Give yourself a tap on the back, you've just completed your first circuit!!!!

Tutorial 2.1a – Walking The Plank

7. Challenges

Well done for completing the tutorial. There's a lot of ground we have covered in this tutorial so please feel free to make notes, come back to the tutorial at some point down the line and ask your learning facilitator any questions or doubts you might have on the concepts covered this far.

For those who want to stretch it a bit further -

1. Sometimes the angle displayed might be negative. Why is that so? Can you work out how to convert the negative number and display it as positive number?
2. Can you change the game so that it includes a timer? Create a time so that a user only has a certain limited time e.g. 30 seconds, to walk the agreed stretch.
3. What happens if you use "rotation(o) pitch" instead of "rotation(o) roll"?

